

Programación Evolutiva

Algoritmo Genético Clásico

Práctica 1

Miguel Márquez Altuna
Fco Javier Álvarez Obeso

Índice

1. Introducción. Algoritmo Genético.....	3
2. Funcionamiento Básico. Resultados para cada función.....	3
2.1. Función 1.....	4
2.2. Función 2.....	4
2.3. Función 3.....	5
2.4. Función 4.....	5
2.5. Función 5.....	6
3. Estudio detallado de los resultados.....	7
3.1. Función 1.....	7
3.1.1. Variación en la probabilidad de cruce.....	7
3.1.2. Selección por ruleta y torneo.....	7
3.2. Función 2.....	8
3.2.1. Variación en el tamaño de la población.....	8
3.2.2. Variación en el número de generaciones.....	9
3.2.3. Variación en la probabilidad de cruce.....	9
3.2.4. Variación en la probabilidad de mutación.....	10
3.2.5. Elitismo.....	11
3.2.6. Selección por ruleta y torneo.....	11
3.3. Función 3.....	12
3.3.1. Variación en el tamaño de la población.....	12
3.4. Función 4.....	13
3.4.1. Variación en el tamaño de la población.....	13
3.4.2. Variación en el número de generaciones.....	13
3.4.3. Variación en la probabilidad de cruce.....	14
3.4.4. Variación en la probabilidad de mutación.....	15
3.4.5. Elitismo.....	16
3.4.6. Selección por ruleta y torneo.....	17
3.5. Función 5.....	18
3.5.1. Variación en el tamaño de la población.....	18
3.5.2. Variación en el número de generaciones.....	20
3.5.3. Variación en la probabilidad de cruce.....	21
3.5.4. Variación en la probabilidad de mutación.....	23
3.5.5. Elitismo.....	25
3.5.6. Selección por ruleta y torneo.....	26
4. Conclusiones.....	27

1. Introducción. Algoritmo Genético

El objetivo de esta práctica es implementar un algoritmo genético clásico para hallar el máximo o mínimo de diferentes funciones.

Para nuestra implementación del algoritmo genético se han seguido las directrices básicas del esquema clásico, con alguna modificación adicional:

Inicialización: Se genera aleatoriamente la población inicial.

Evaluación: A cada uno de los cromosomas de esta población se aplica la función de adaptación.

Bucle: A continuación se itera, tantas veces como se especifique por parámetro los siguientes pasos:

- **Selección** Se genera la población intermedia media una *ruleta clásica* o mediante *torneo determinista* sobre grupos de 3 individuos.
- **Reproducción** Para cada cromosoma de la población intermedia se van seleccionando aquellos que se van a reproducir en base a la probabilidad de cruce. Una vez seleccionados se toman dos a dos de forma arbitraria, se realiza cruce monopunto y se emplea un mecanismo de *reemplazamiento por inclusión* en el que mezclan padres e hijos en una misma población para posteriormente seleccionar los mejores.
- **Mutación** básica que modifica al azar parte del cromosoma de los individuos.
- **Evaluación.** Se realiza distinción entre función de evaluación y de adaptación. Sobre los valores de la función de evaluación se aplica una función de desplazamiento para garantizar el correcto funcionamiento de los mecanismos de selección.

En el caso elitismo se implementó de forma simple, se realiza la separación previa a la etapa de selección del porcentaje de individuos de la población que se desee conservar como élite y se vuelve a unir a la población tras la mutación.

2. Funcionamiento básico. Resultados para cada función

Introducidas las características de la implementación, procedemos a mostrar los resultados de la misma, tomada la siguiente configuración de parámetros como referencia:

Tamaño Población - 100

Número de generaciones - 100

Probabilidad de cruce - 0.6

Probabilidad de mutación - 0.05

Tolerancia - 0.0001

Selección - Ruleta

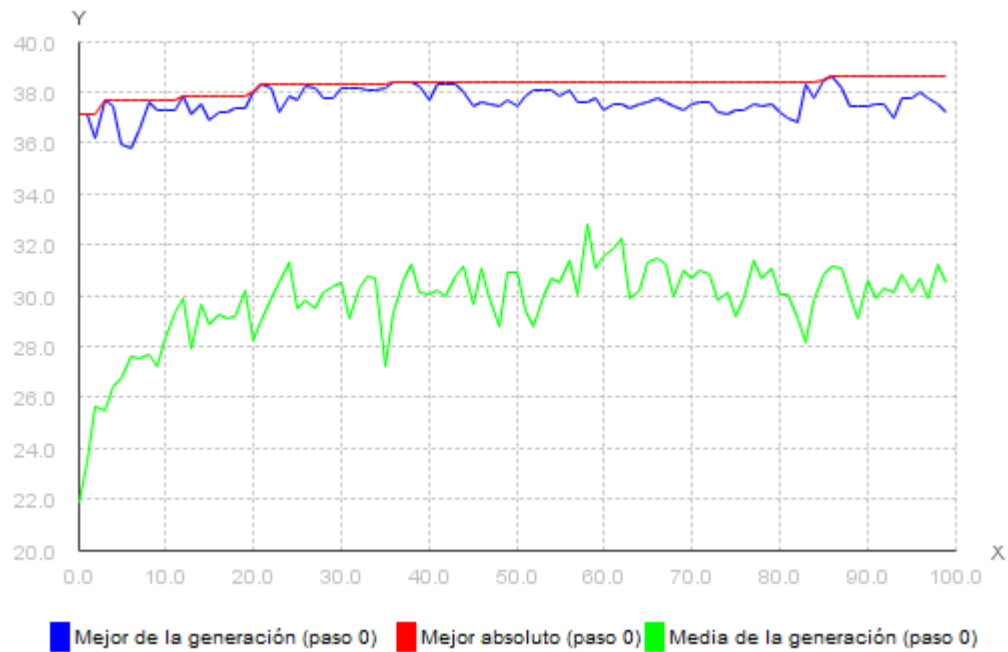
Elitismo - No

Se tienen los siguientes resultados:

Programación Evolutiva - Práctica 1

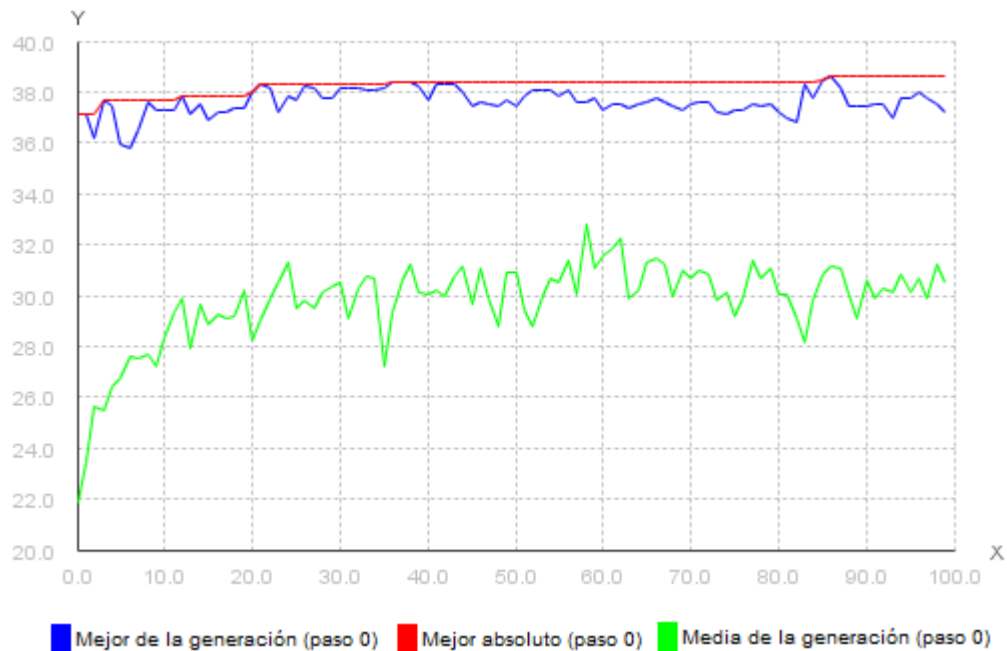
Algoritmo Genético Clásico

2.1. Función 1



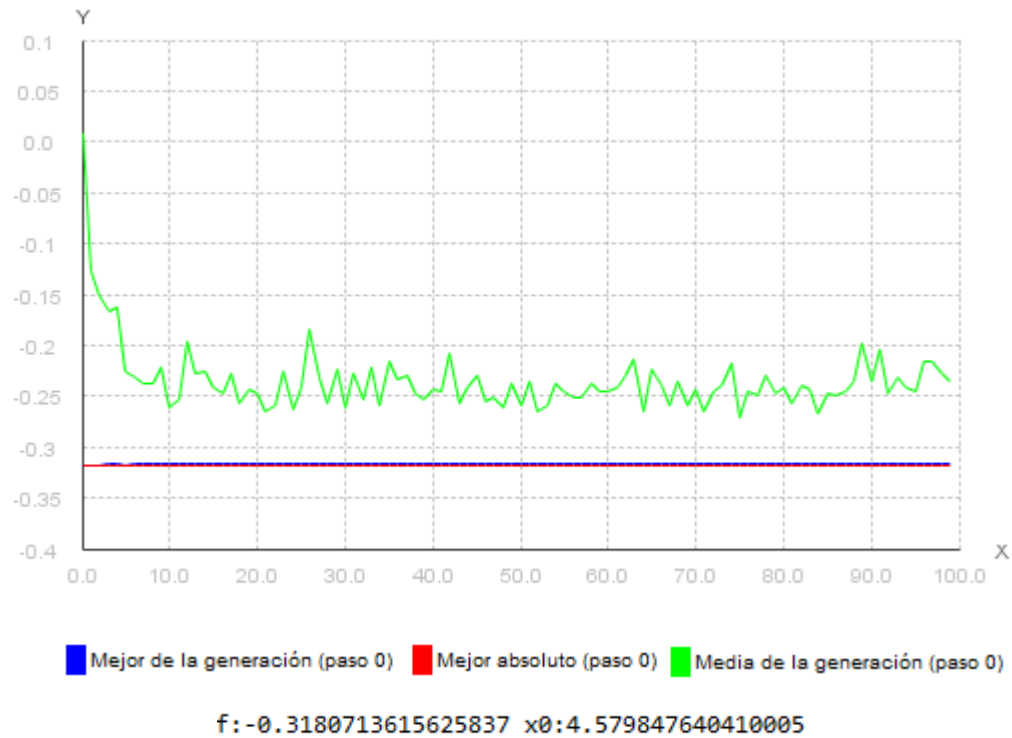
f:1.9844219887442724 x0:0.9844961240310077

2.2. Función 2

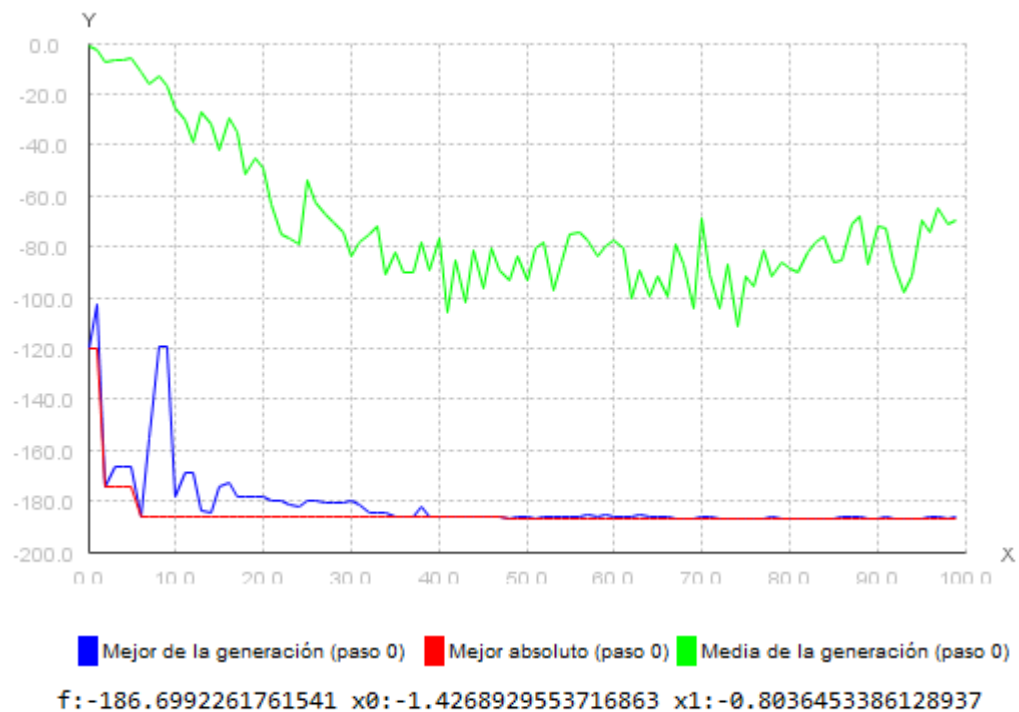


f:38.79199493902921 x0:11.625127888213685 x1:5.727314065981018

2.3. Función 3



2.4. Función 4

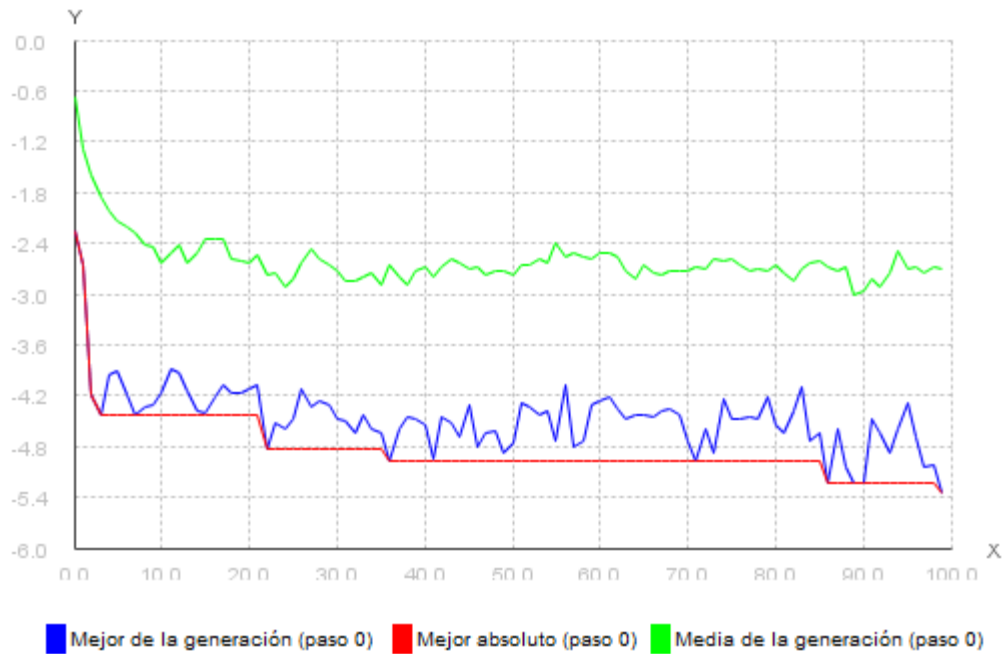


Programación Evolutiva - Práctica 1

Algoritmo Genético Clásico

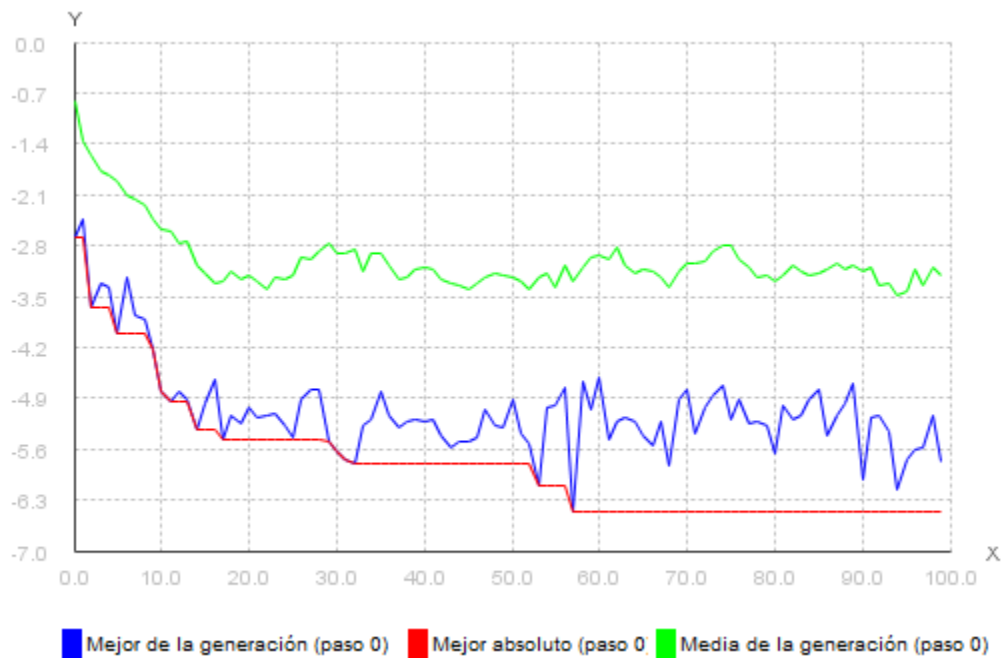
2.5. Función 5

Para n = 6:



f: -5.513218851547402 x0:1.595580460250903 x1:1.2874326655599762 x2:1.0741069520605016
x3:1.7263563133804687 x4:1.5689267306541146 x5:1.8619260027684494

Para n = 8:



f: -6.371658884810858 x0:1.5613524693658187 x1:1.2887749397123325 x2:1.0918441462166377 x3:1.7679668121035124
x4:1.533931725967684 x5:1.884936416808842 x6:1.3495607834690369 x7:0.7666302944457529

3. Estudio detallado de cada función.

Conocidos los resultados, a continuación partiendo de la misma población inicial en todos los estudios, para cada caso se irá modificando uno de los parámetros y se analizarán los efectos sobre el problema. Se consideran los siguientes valores fijos de parametrización:

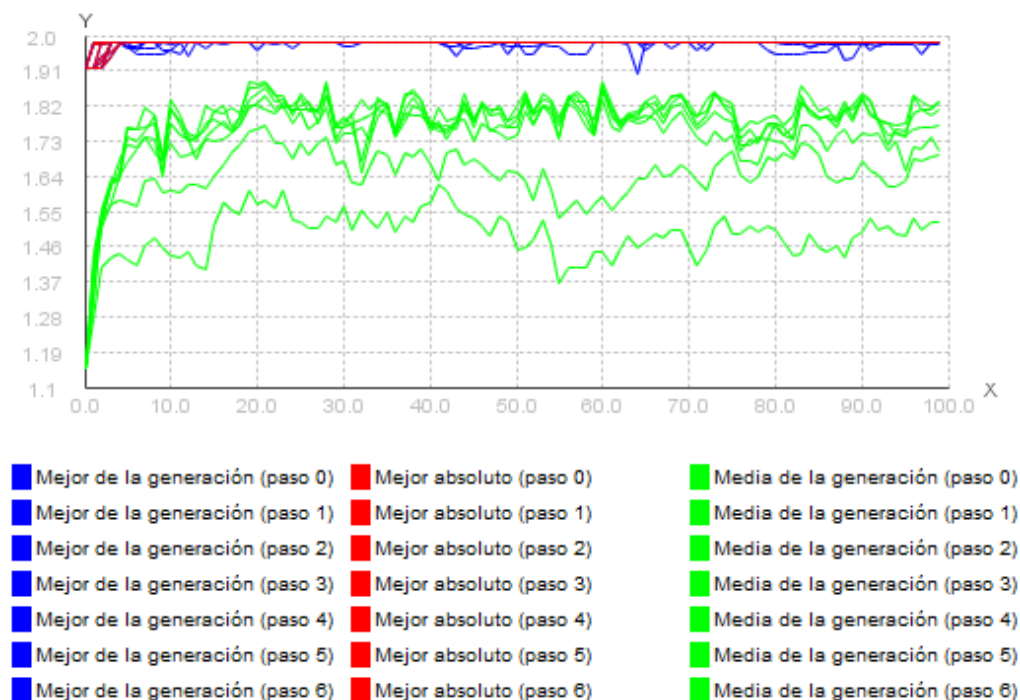
Tamaño Población - 100
Número de generaciones - 100
Probabilidad de cruce - 0.6
Probabilidad de mutación - 0.05
Tolerancia - 0,0001
Selección - Ruleta
Elitismo - No

3.1. Función 1

En términos de convergencia esta función, es la más rápida de todas, independientemente de la configuración de parámetros siempre existe una convergencia por debajo de la 10ª generación. La media de la población, en cambio, si se ve afectada por ello. Sólo se ha estudiado el único parámetro que provoca algún cambio significativo, la probabilidad de cruce.

3.1.1. Variación de la probabilidad de cruce

Tomando valores entre 0 y 100% en incrementos de un 20% se tiene el siguiente resultado:



f:1.9844219887442724 x0:0.9844961240310077

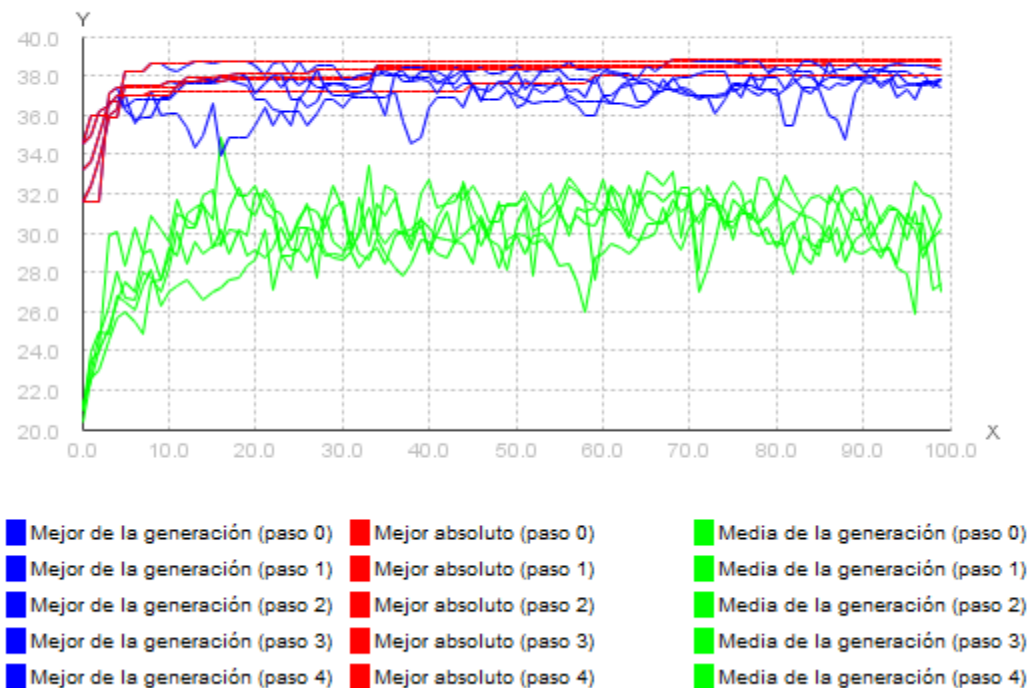
A medida que se aumenta la probabilidad de cruce la media mejora, hasta cierto punto a partir del 60% que se alcanza un valor estable. Esto se debe en parte al uso de un mecanismo de reemplazamiento inclusivo que tras cada cruce hace que sólo se conserven los nuevos individuos que al menos fueran mejores que los peores de la generación anterior.

3.2. Función 2

Esta función requiere de un número mayor de generaciones que la anterior para conseguir valores máximos cercanos al máximo real, aunque no exige que este número sea demasiado alto, ya que converge rápidamente.

3.2.1. Variación en el tamaño de la población

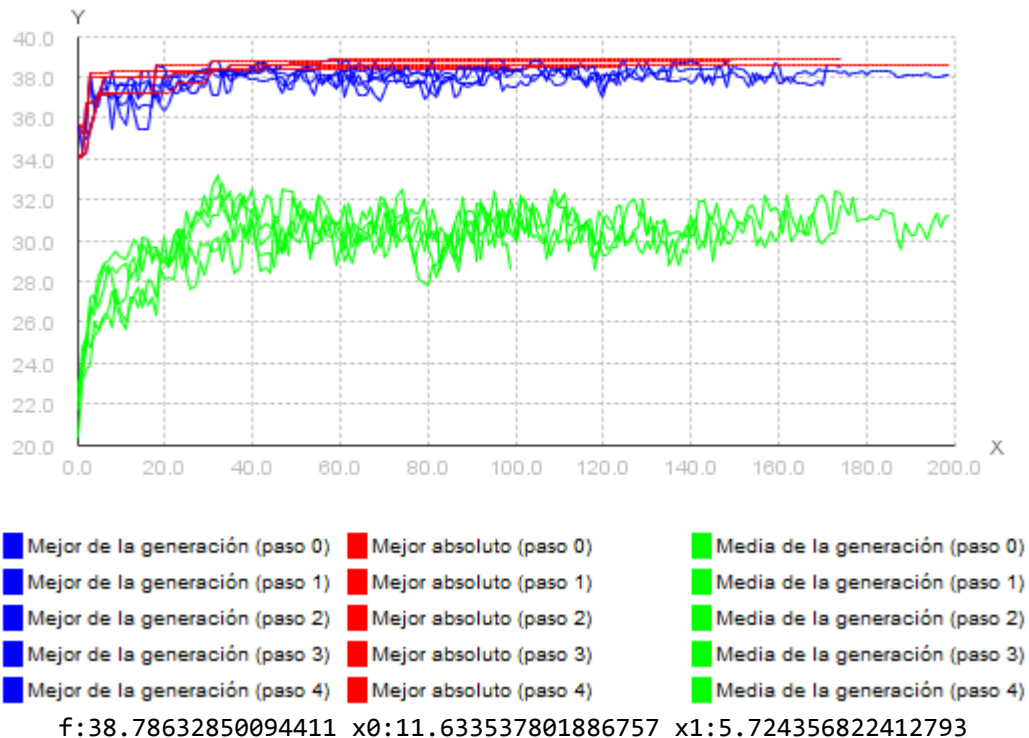
Variando el tamaño de la población de 20 a 100 individuos en incrementos de 20 individuos se puede comprobar que este parámetro no afecta demasiado a esta función, obteniendo resultados similares. Sin embargo, sí que se observa que no existe una proporción directa entre el tamaño de la población y la calidad del máximo obtenido (en muchos casos, con poblaciones más pequeñas se consiguen mejores resultados). Esto puede ser debido a que una población de más individuos introduce ruido, retrasando la convergencia.



f:38.78160543042683 x0:11.61810042610331 x1:5.723786126285592

3.2.2. Variación en el número de generaciones

En este caso se puede comprobar cómo la mejora no es demasiado significativa, variando el número de generaciones desde 100 a 200: la función alcanza un valor cercano al máximo relativamente pronto (alrededor de la 10ª generación), con un par de mejoras en generaciones posteriores, mucho antes de las últimas generaciones, que no aportan mejoras.

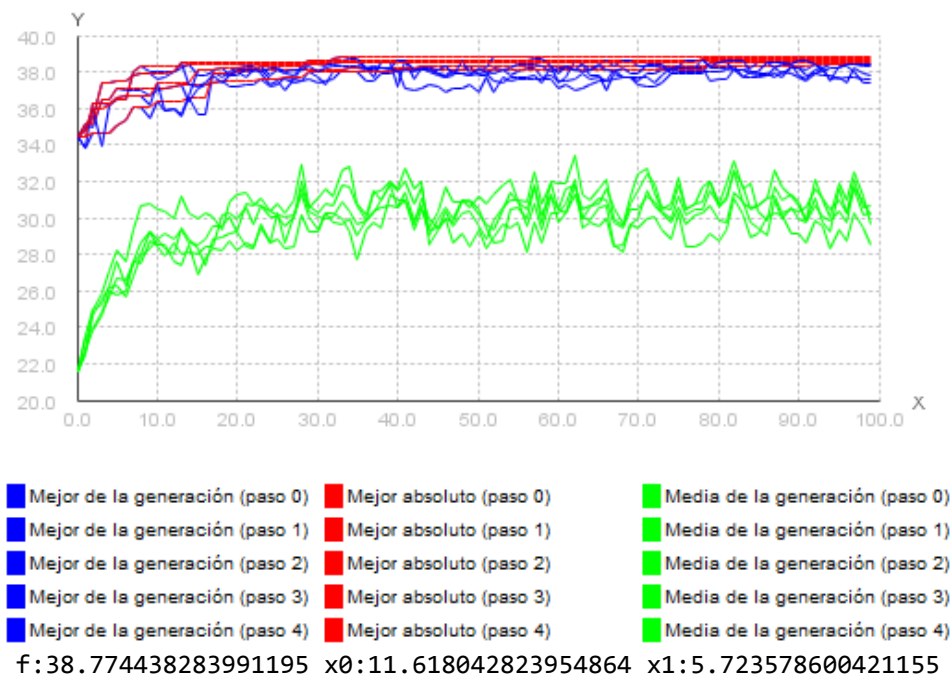


3.2.3. Variación en la probabilidad de cruce

Variando la probabilidad de cruce constatamos lo que ya suponíamos, los mejores resultados los obtenemos con la probabilidad de cruce alrededor del 60%, mientras que valores menores o mayores ofrecen resultados más pobres. En esta simulación, concretamente, hemos variado la probabilidad de cruce entre el 40 y el 80% con incrementos del 10%

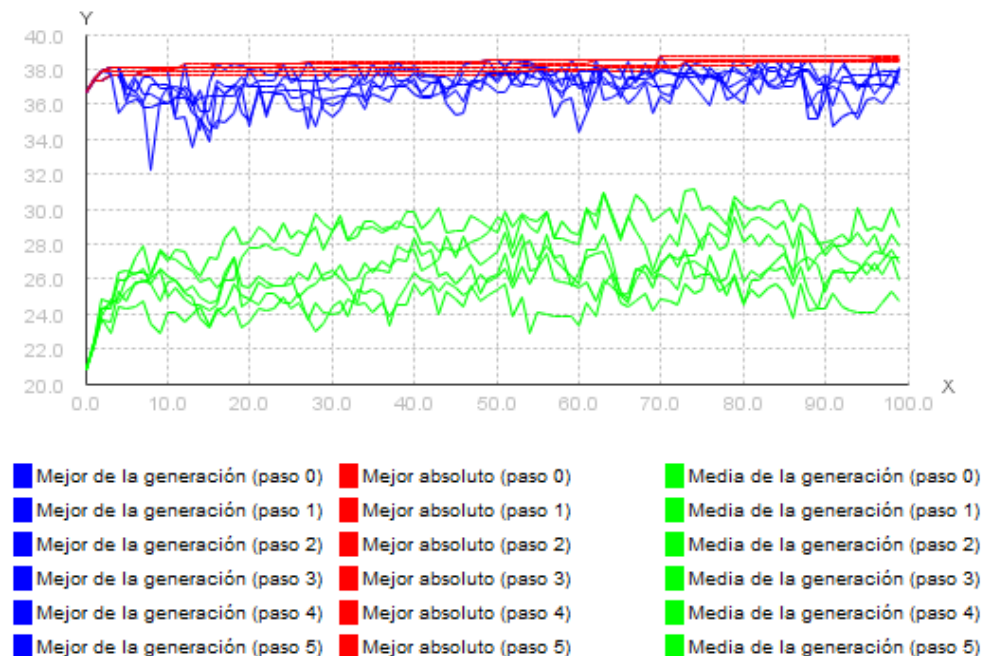
Programación Evolutiva - Práctica 1

Algoritmo Genético Clásico



3.2.4. Variación en la probabilidad de mutación

Mediante la variación de la probabilidad de mutación, entre un 5 y un 10%, no se observan grandes diferencias en el cálculo del máximo de la función (que puede ser debido a que la función es relativamente sencilla). Sin embargo, sí que se aprecian variaciones importantes tanto en la aptitud del mejor individuo de cada población como en la aptitud del individuo medio.



3.2.5. Elitismo

Añadiendo elitismo podemos observar cómo la convergencia al máximo es más rápida, aparte de que el mejor individuo de cada generación coincide con el mejor individuo hasta el momento.

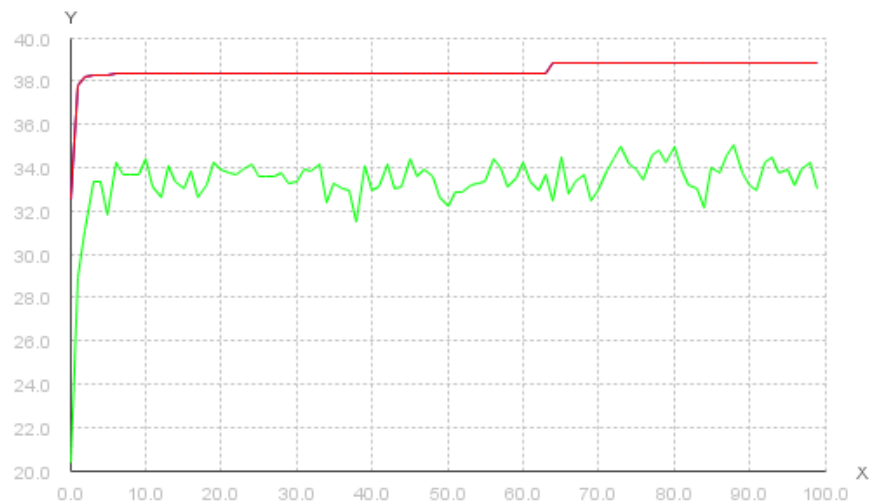


■ Mejor de la generación (paso 0) ■ Mejor absoluto (paso 0)
■ Media de la generación (paso 0)

f:38.79815862084945 x0:11.61971328625979 x1:5.723682363353373

3.2.6. Selección por ruleta y torneo

En este caso podemos ver que el torneo como método de selección ofrece mejores resultados para esta función que la ruleta: converge más rápidamente y hace mejorar la aptitud media de la población.



■ Mejor de la generación (paso 0) ■ Mejor absoluto (paso 0)
■ Media de la generación (paso 0)

f:38.747427187266354 x0:11.625588705401249 x1:5.7220221564378795

3.3. Función 3

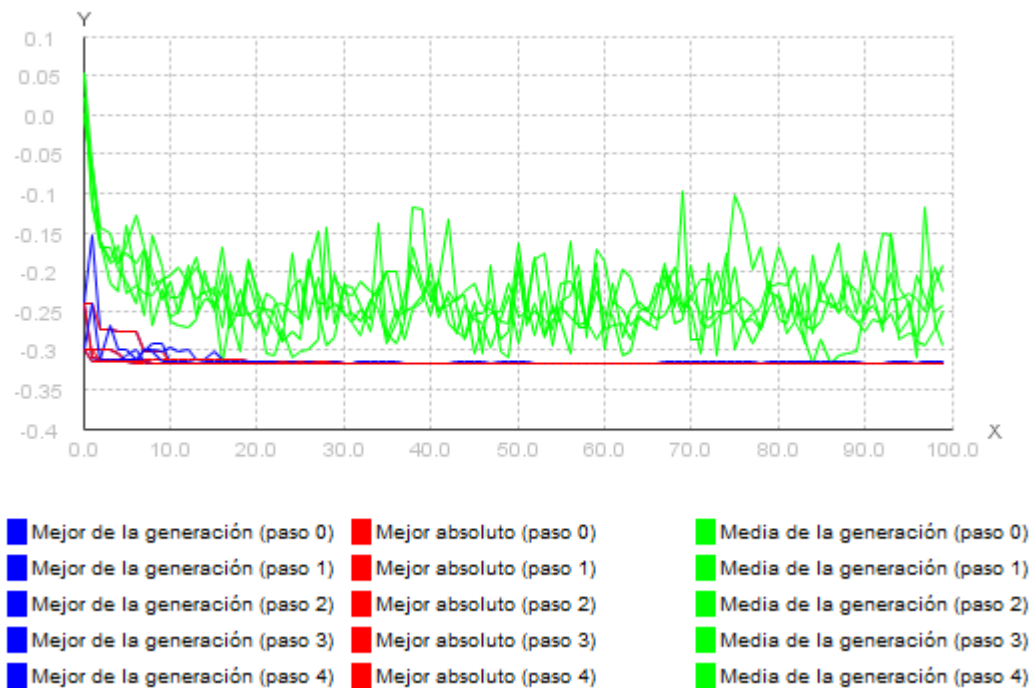
A diferencia de las anteriores, el objetivo para esta función es encontrar el mínimo. Aparentemente, se trata de una función bastante sencilla de minimizar, pues para casi cualquier configuración de los parámetros se obtiene una rápida convergencia al valor mínimo.

Intuitivamente, algunos individuos de la población aleatoria inicial adquieren valores muy cercanos al mínimo que estamos buscando, lo que provoca la rápida convergencia. Así pues, nos interesará variar el tamaño de la población para comparar la velocidad de convergencia.

El resto de parámetros no ofrecen cambios relevantes, ni siquiera el cambio del método de selección de ruleta por torneo.

3.3.1. Variación en el tamaño de la población

Queremos forzar un tamaño pequeño de población inicial para comprobar cómo influye el tamaño de la población inicial en la velocidad de convergencia. Así pues, variaremos el tamaño de la población de 10 a 50 individuos, con incrementos de 10.



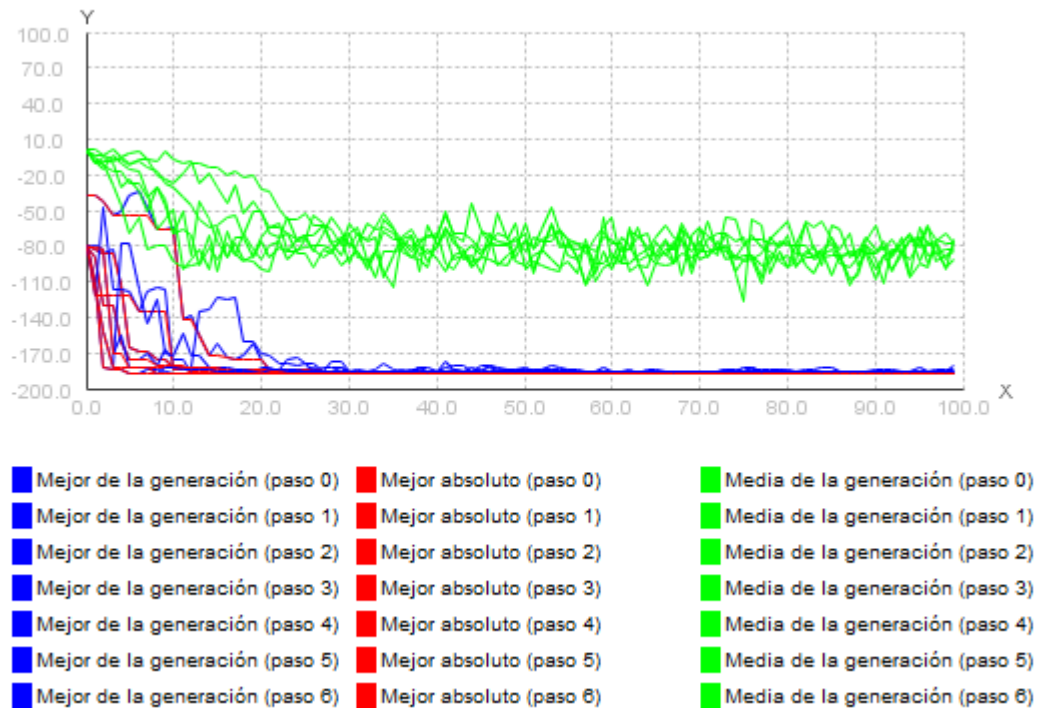
f: -0.3180713615625837 x0:4.579847640410005

Sin embargo, a pesar de reducir drásticamente el tamaño de la población y ver, efectivamente, cómo se reduce la velocidad de convergencia, esta sigue siendo muy rápida, puesto que en el peor de los casos alcanza el valor mínimo antes de la décima generación.

3.4. Función 4

3.4.1. Variación del tamaño de la población

Tomando valores desde 20 hasta 100 con paso 13 se tiene:



f: -186.73023782587245 x0: -7.708655199642942 x1: -0.8007461576315222

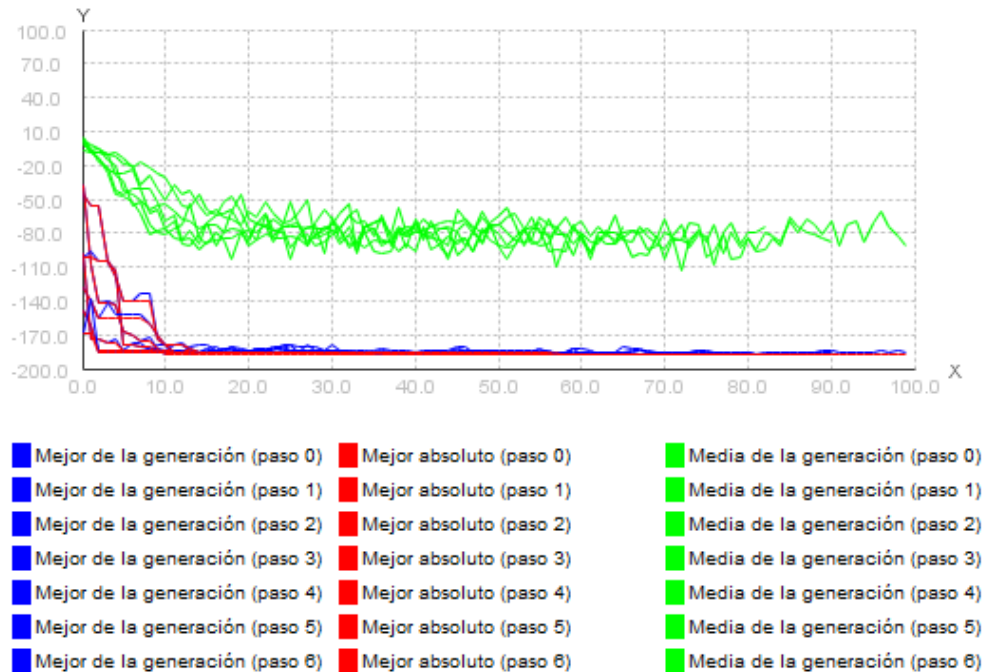
Aunque no se aprecian cambios significativos en la media, vemos que cuanto más pequeña es la población más tarda la función en converger, como en el caso de una población de tamaño 20, que hasta la generación número 20 no se estabiliza.

3.4.2. Variación del número de generaciones

Tomando valores desde 20 hasta 100 con paso 13 se tiene:

Programación Evolutiva - Práctica 1

Algoritmo Genético Clásico



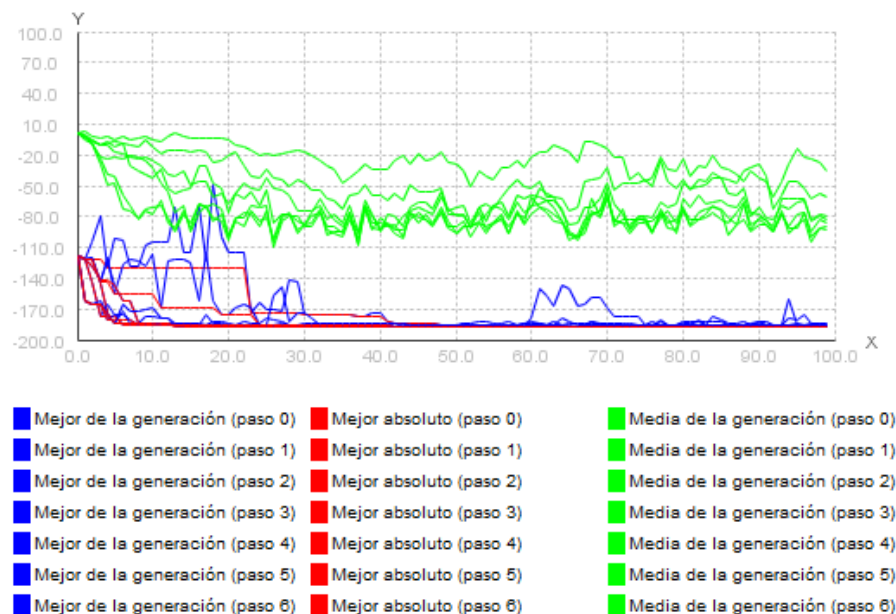
f: -186.6992261761541 x0: -1.4268929553716863 x1: -0.8036453386128937

Vemos que la convergencia de esta función es rápida más allá de la generación 10, el algoritmo ya se estabiliza y alcanza la mejor solución, aunque no se aprecie en los gráficos la mejor solución siempre es alcanzada por aquella ejecución con más generaciones.

Ha de notarse que para esta prueba no se ha utilizado la misma secuencia de números aleatorios entre pasos.

3.4.3. Variación de la probabilidad de cruce

Tomando valores entre 0 y 100% en incrementos de un 20% se tiene el siguiente resultado:

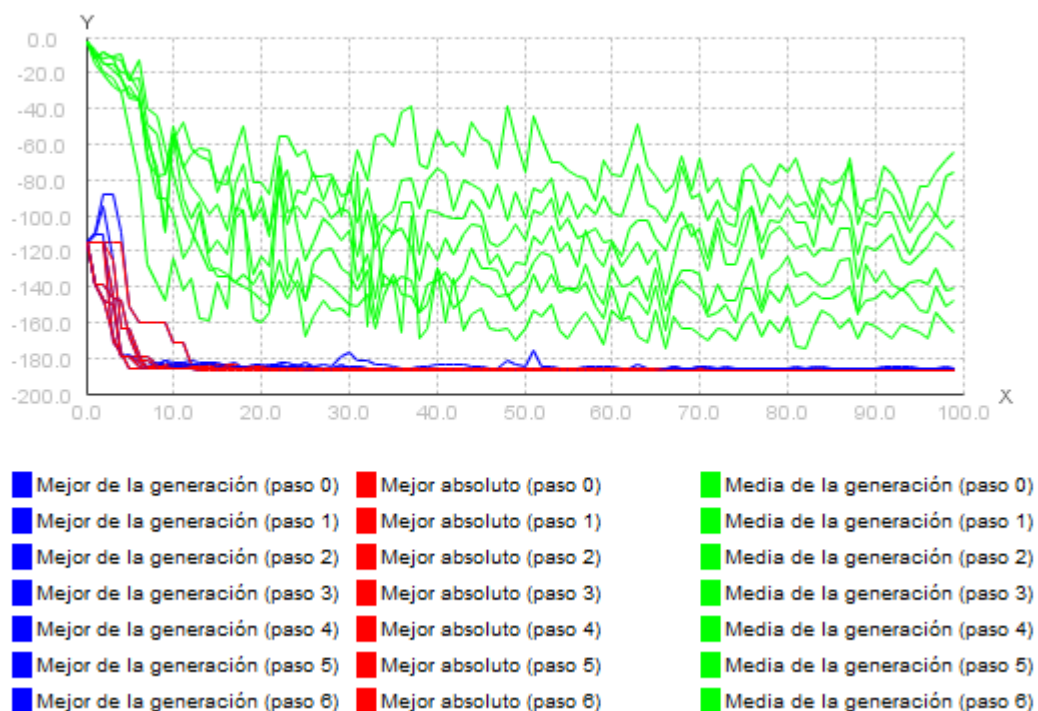


f: -186.73071007058212 x0: -1.4249855994628877 x1: -0.8000595095043543

A medida que la probabilidad de cruce es mayor, la media de la población mejora y la convergencia es más rápida, ya que existe una mayor posibilidad de mejora; además, al usar reemplazamiento inclusivo no se puede empeorar con el cruce por lo que todo incremento de este parámetro mejora los resultados.

3.4.4. Variación de la probabilidad de mutación

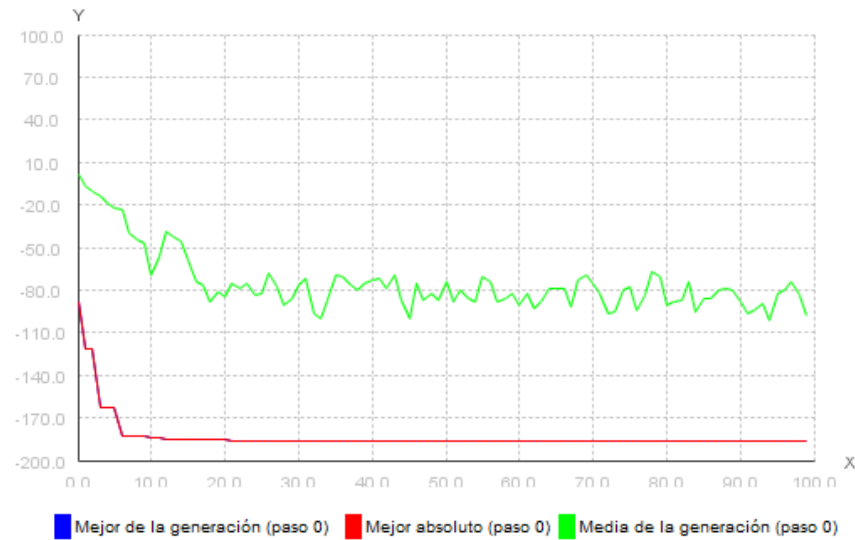
Considerados valores desde el 1% hasta el 5% con un paso de 1% obtenemos el siguiente resultado:



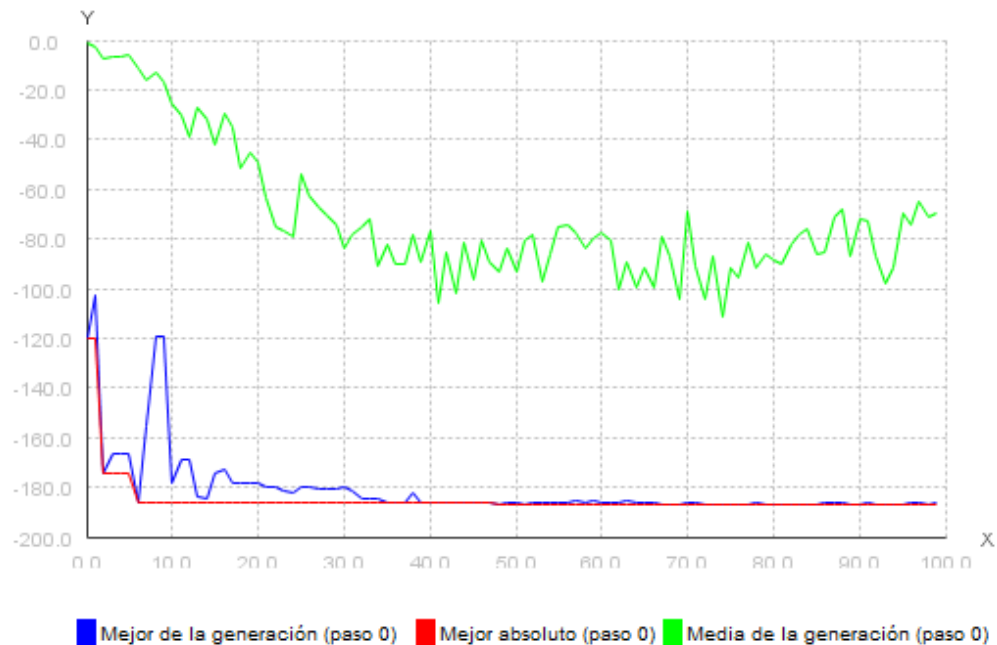
f: -186.72790558513293 x0: -7.708655199642942 x1: -0.8014328057586884

Donde podemos observar que a medida que vamos aumentando la probabilidad de cruce la aptitud del individuo medio empeora hasta que se alcanza un valor entorno al 5% donde se estabiliza. Así mismo la convergencia es más rápida conforme se aumenta, puesto que esta contribuyendo a explorar otros valores en el espacio de búsqueda.

3.4.5. Elitismo



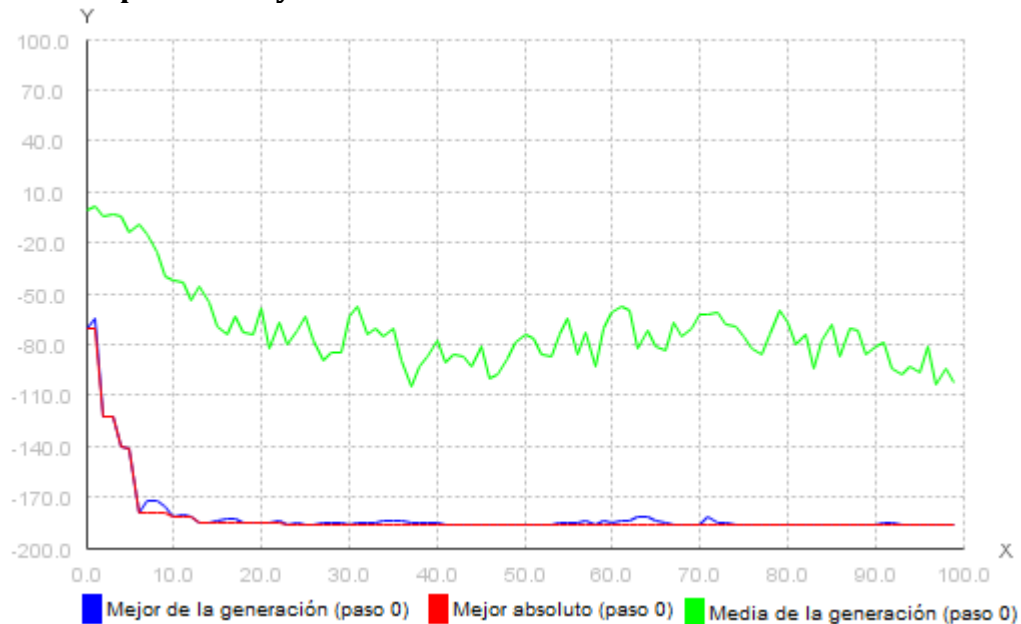
f: -186.70527401866113 x0: 4.8613542989894825 x1: -0.8006698633951697



f: -186.72788151358185 x0: -7.708578905406592 x1: -7.084644640520632

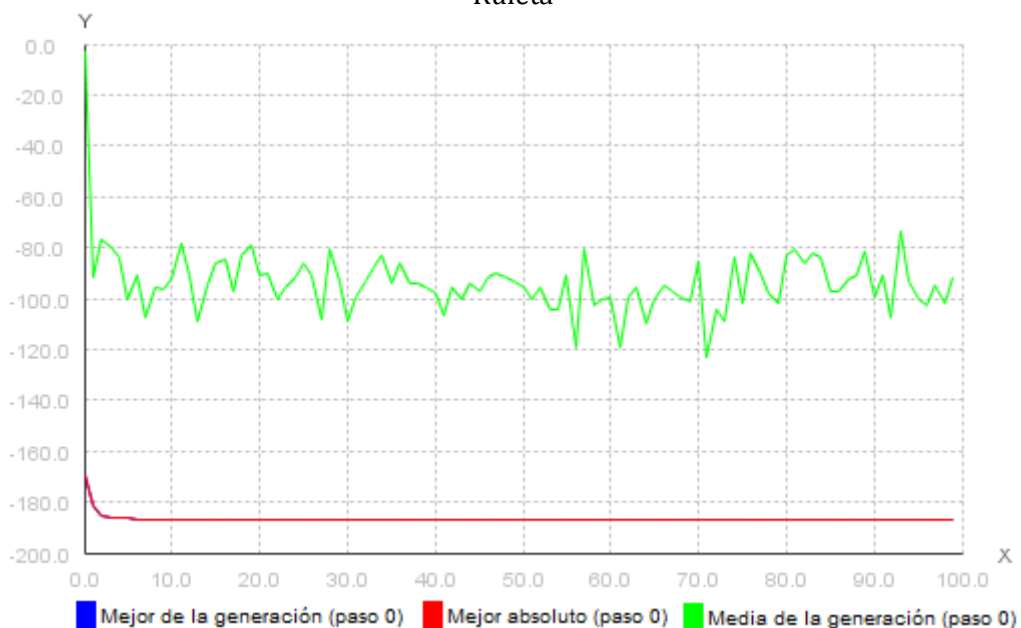
Podemos intuir como se conserva ese conjunto de super individuos (2% de los mejores de la población) ya que el mejor de cada generación coincide siempre con el mejor absoluto.

3.4.6. Selección por ruleta y torneo



f: -186.71993755667435 x0: -0.8025009250676156 x1: -1.424680422517481

Ruleta



f: -186.73035027580934 x0: -7.708350022697536 x1: -0.800822451867873

Torneo

En este caso ambos métodos de selección nos permiten obtener el mejor resultado y no se aprecian diferencias significativas en lo que se refiere al resultado o la media. Con la selección por torneo vemos que siempre el mejor de la generación coincide con el mejor absoluto, puesto que en las agrupaciones el mejor de la generación se impone en el “torneo” y además al emplear reemplazamiento por inclusión sólo podría verse reemplazado si se han generado individuos mejores en el cruce.

3.5. Función 5

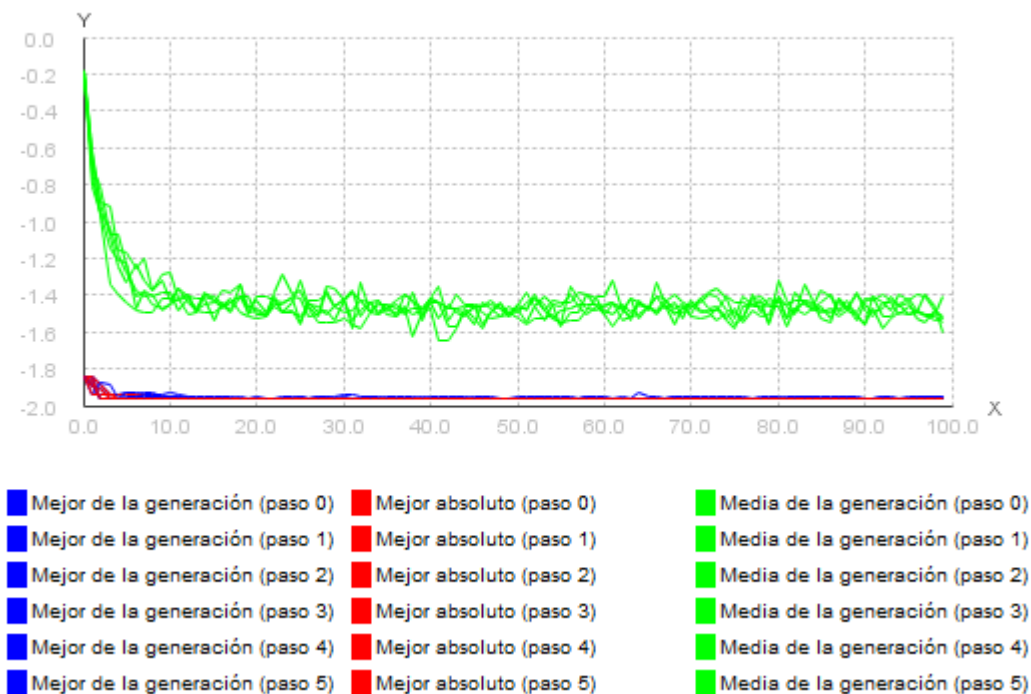
Se trata de la función más compleja de las que se proponen, cuya complejidad, además, aumenta según el número de variables que fijemos. Al igual que las dos anteriores, el objetivo para esta es la minimización, y obtendremos resultados muy diversos según los parámetros que elijamos.

3.5.1. Variación en el tamaño de la población

Al igual que con las funciones anteriores, si bien el aumento del tamaño de la población consigue, por lo general, acercar el valor obtenido al mínimo real, no existe una relación directa que refleje que a mayor tamaño de población, mejores resultados. De igual forma, como se ha ido viendo con anterioridad, podemos ver cómo el aumento del tamaño de la población mejora la velocidad de convergencia.

En este caso, hemos variado el tamaño de la población de 100 a 200 individuos, con incrementos de 20 individuos, para unos valores de $n = 2, 5$ y 7 .

Con $n = 2$:

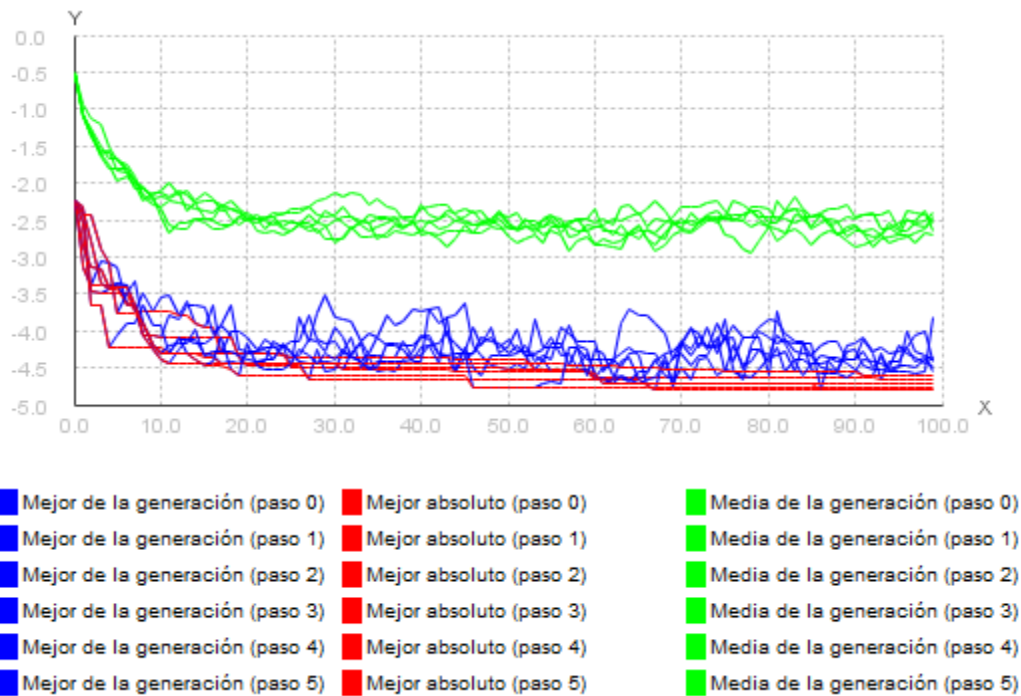


f:-1.959089286576618 x0:1.5709401418826492 x1:1.285131624155937

Programación Evolutiva - Práctica 1

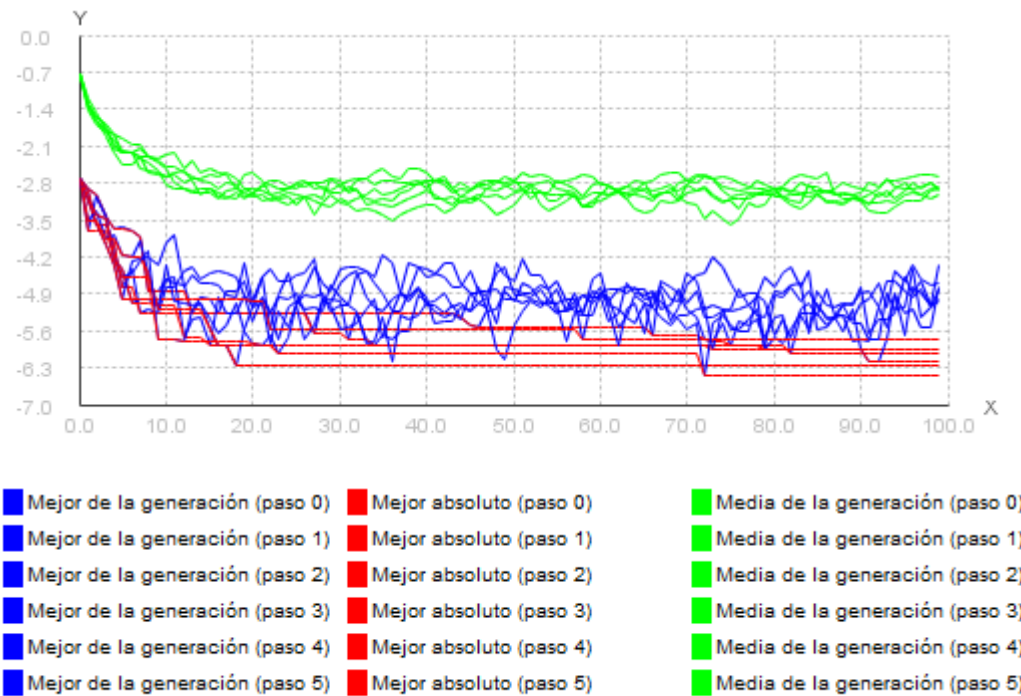
Algoritmo Genético Clásico

Con n = 5:



f: -4.795416301689821 x0: 1.6039217353405453 x1: 1.2764068421656214
x2: 1.9263551620815493 x3: 1.7089067493998376 x4: 1.5725700462105103

Con n = 7:

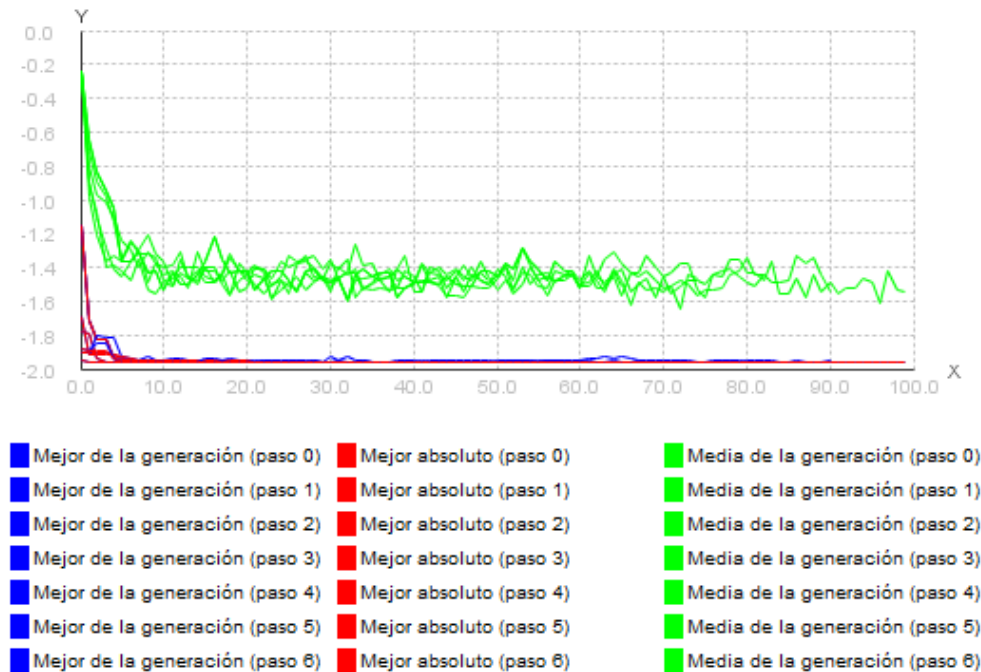


f: -6.448597661836693 x0: 1.5960598438767444 x1: 1.2914594880170451
x2: 1.1288525621316028 x3: 1.0127458479527875 x4: 1.5742958272635394 x5: 1.4352745757695
x6: 1.357230921482501

3.5.2. Variación en el número de generaciones

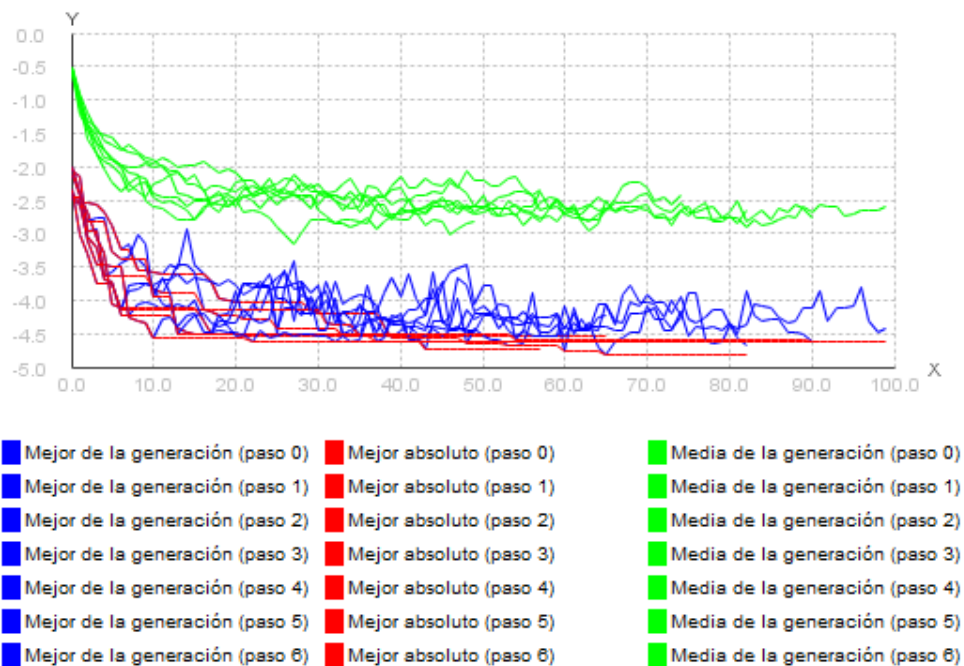
Tomando valores desde 20 hasta 100 con paso 13 se tiene:

Para n = 2:



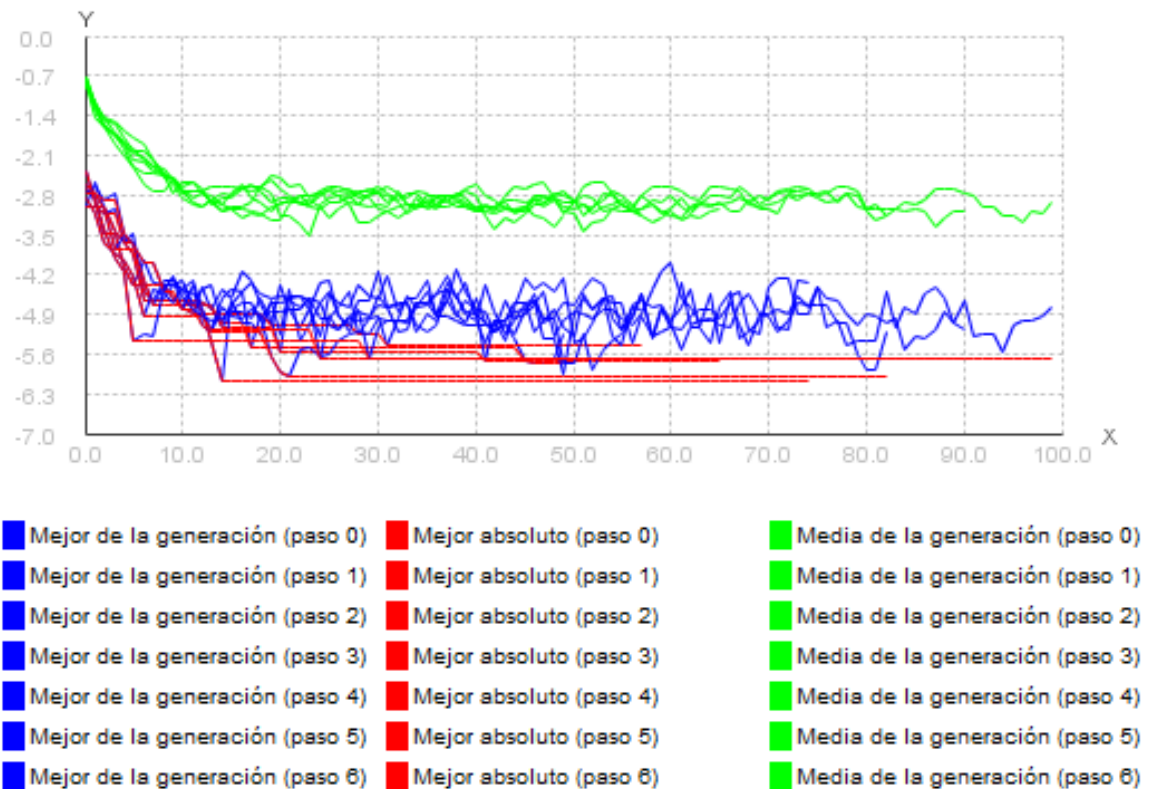
f: -1.959064290589058 x0:1.5711318953329856 x1:1.2856110077817786

Para n = 5:



f: -4.816890022407064 x0:1.5583802908856013 x1:1.2529170444993871
x2:1.9281768198597469 x3:1.7194531891683507 x4:1.5715154022336588

Para $n = 7$:



f: -6.0798656988354605 x0:1.5057439687682028 x1:1.2646140049699202
x2:2.4844535792862423 x3:1.703058269164571 x4:1.5718030324091636 x5:1.871321921834943
x6:2.072950674863885

Donde vemos que para mayores valores n , el número de generaciones tampoco es indicativo de que vayamos a obtener el mejor resultado en una ejecución. Ha de notarse que para esta prueba no se ha utilizado la misma secuencia de números aleatorios entre pasos.

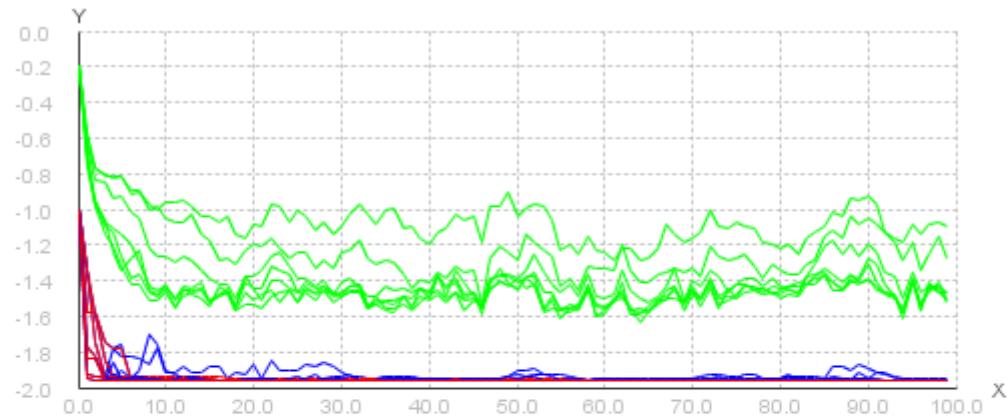
3.5.3. Variación en la probabilidad de cruce

Tomando valores entre 0 y 100% en incrementos de un 20% obtenemos los siguientes resultados:

Con $n = 2$:

Programación Evolutiva - Práctica 1

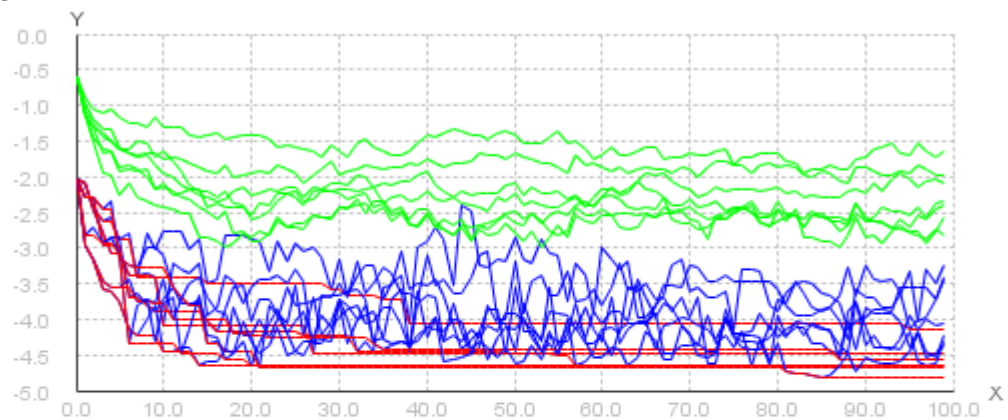
Algoritmo Genético Clásico



■ Mejor de la generación (paso 0)	■ Mejor absoluto (paso 0)	■ Media de la generación (paso 0)
■ Mejor de la generación (paso 1)	■ Mejor absoluto (paso 1)	■ Media de la generación (paso 1)
■ Mejor de la generación (paso 2)	■ Mejor absoluto (paso 2)	■ Media de la generación (paso 2)
■ Mejor de la generación (paso 3)	■ Mejor absoluto (paso 3)	■ Media de la generación (paso 3)
■ Mejor de la generación (paso 4)	■ Mejor absoluto (paso 4)	■ Media de la generación (paso 4)
■ Mejor de la generación (paso 5)	■ Mejor absoluto (paso 5)	■ Media de la generación (paso 5)
■ Mejor de la generación (paso 6)	■ Mejor absoluto (paso 6)	■ Media de la generación (paso 6)

f: -1.959088787353756 x0:1.5705566349819757 x1:1.2849398707056003

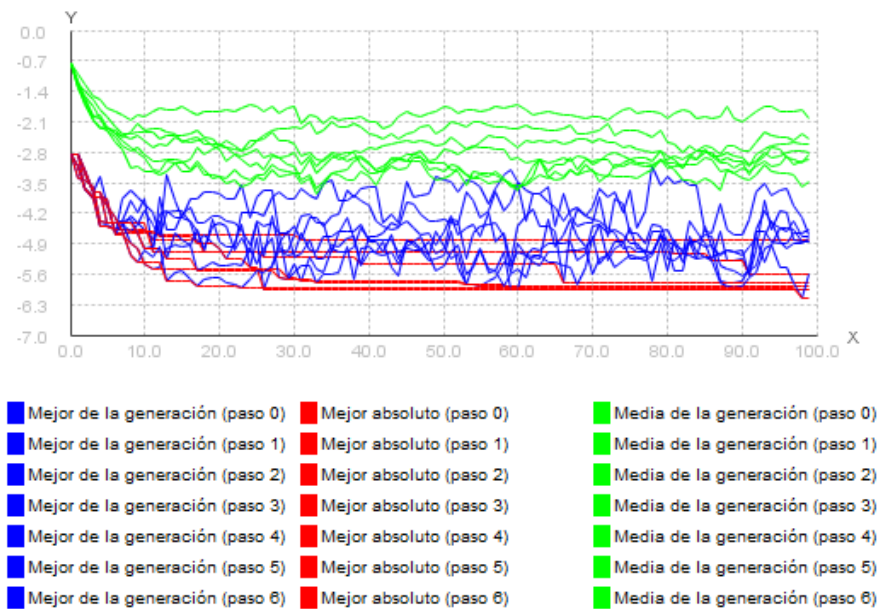
Para n = 5:



■ Mejor de la generación (paso 0)	■ Mejor absoluto (paso 0)	■ Media de la generación (paso 0)
■ Mejor de la generación (paso 1)	■ Mejor absoluto (paso 1)	■ Media de la generación (paso 1)
■ Mejor de la generación (paso 2)	■ Mejor absoluto (paso 2)	■ Media de la generación (paso 2)
■ Mejor de la generación (paso 3)	■ Mejor absoluto (paso 3)	■ Media de la generación (paso 3)
■ Mejor de la generación (paso 4)	■ Mejor absoluto (paso 4)	■ Media de la generación (paso 4)
■ Mejor de la generación (paso 5)	■ Mejor absoluto (paso 5)	■ Media de la generación (paso 5)
■ Mejor de la generación (paso 6)	■ Mejor absoluto (paso 6)	■ Media de la generación (paso 6)

f: -4.726985829542572 x0:1.5696937444554613 x1:1.2479314547906353
x2:1.1347969190920375 x3:1.715713996886787 x4:1.5740081970880346

Para $n = 7$:



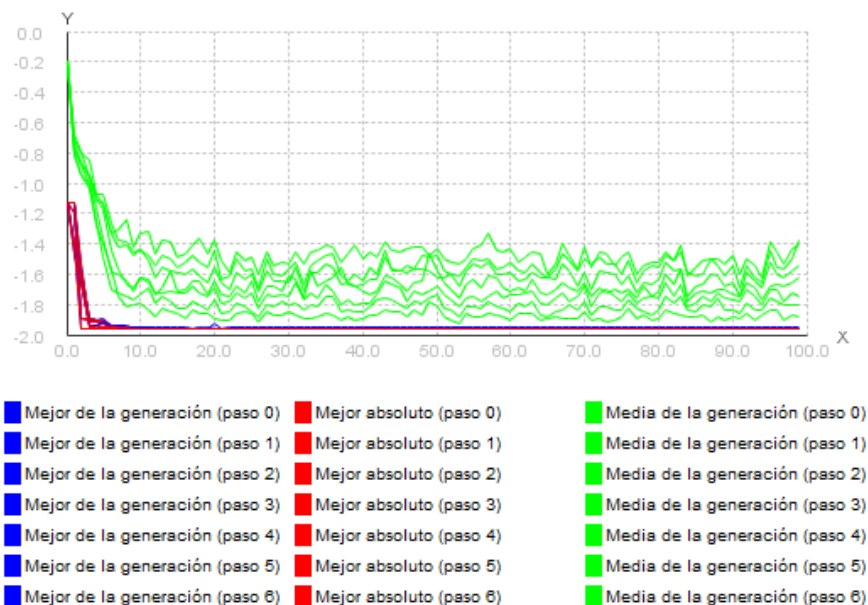
f: -6.213139603427407 x0:1.5174409292387359 x1:1.2394943029758247 x2:1.129236069032276
x3:1.7155222434364503 x4:1.5543534684285325 x5:0.84246878405388 x6:1.3674897310755096

Para $n = 2$ sólo se ve afectado el valor de la media de cada generación, pero para valores mayores valores de n , obtenemos mejor el resultado general así como la media cuanto mayor sea la probabilidad de cruce

3.5.4. Variación en la probabilidad de mutación

Realizada una variación de los parámetros entre el 1% y el 10% con 6 pasos intermedios se tienen los siguientes resultados:

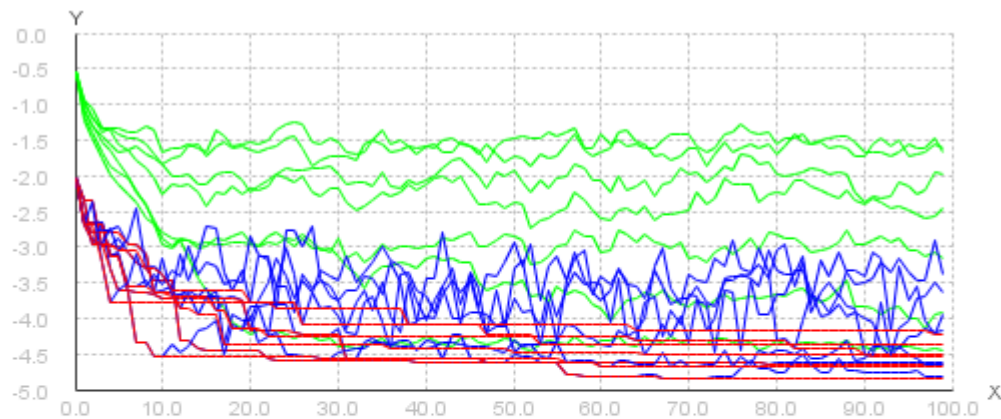
Para $n = 2$:



f: -1.9590910628484648 x0:1.5708442651574808 x1:1.2850357474307688

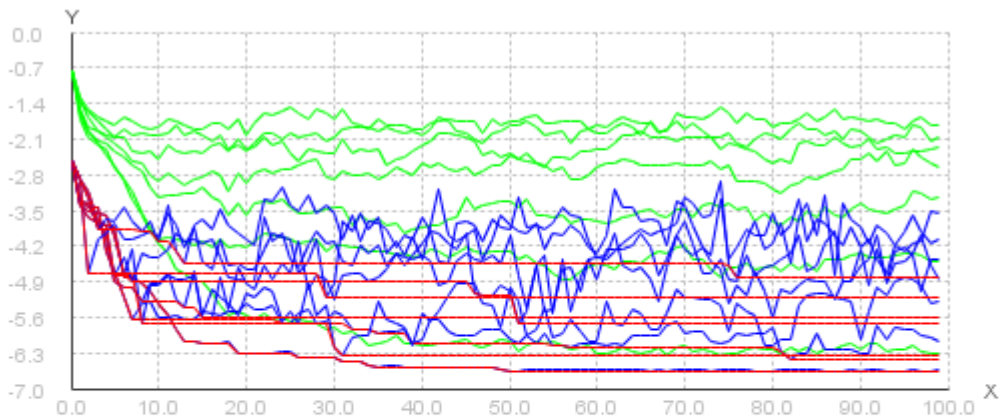
Programación Evolutiva - Práctica 1

Algoritmo Genético Clásico



■ Mejor de la generación (paso 0)	■ Mejor absoluto (paso 0)	■ Media de la generación (paso 0)
■ Mejor de la generación (paso 1)	■ Mejor absoluto (paso 1)	■ Media de la generación (paso 1)
■ Mejor de la generación (paso 2)	■ Mejor absoluto (paso 2)	■ Media de la generación (paso 2)
■ Mejor de la generación (paso 3)	■ Mejor absoluto (paso 3)	■ Media de la generación (paso 3)
■ Mejor de la generación (paso 4)	■ Mejor absoluto (paso 4)	■ Media de la generación (paso 4)
■ Mejor de la generación (paso 5)	■ Mejor absoluto (paso 5)	■ Media de la generación (paso 5)
■ Mejor de la generación (paso 6)	■ Mejor absoluto (paso 6)	■ Media de la generación (paso 6)

f: -4.795325448657628 x0:1.5859927877340725 x1:1.3077585312956566
 x2:1.1102524774489517 x3:1.7246305323274391
 x4:1.5676803332269267



■ Mejor de la generación (paso 0)	■ Mejor absoluto (paso 0)	■ Media de la generación (paso 0)
■ Mejor de la generación (paso 1)	■ Mejor absoluto (paso 1)	■ Media de la generación (paso 1)
■ Mejor de la generación (paso 2)	■ Mejor absoluto (paso 2)	■ Media de la generación (paso 2)
■ Mejor de la generación (paso 3)	■ Mejor absoluto (paso 3)	■ Media de la generación (paso 3)
■ Mejor de la generación (paso 4)	■ Mejor absoluto (paso 4)	■ Media de la generación (paso 4)
■ Mejor de la generación (paso 5)	■ Mejor absoluto (paso 5)	■ Media de la generación (paso 5)
■ Mejor de la generación (paso 6)	■ Mejor absoluto (paso 6)	■ Media de la generación (paso 6)

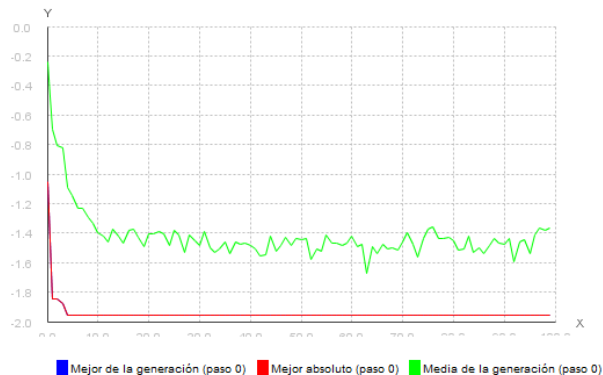
f: -6.564379168555054 x0:1.58388349978037 x1:2.205931692672322 x2:1.1108277377999616
 x3:1.71667276413847 x4:1.565475168548056 x5:1.8773621555205464 x6:1.7589543999376915

Programación Evolutiva - Práctica 1

Algoritmo Genético Clásico

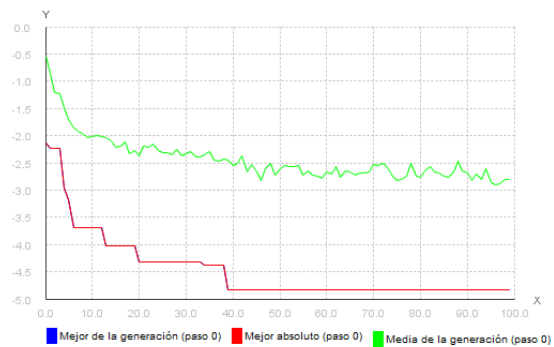
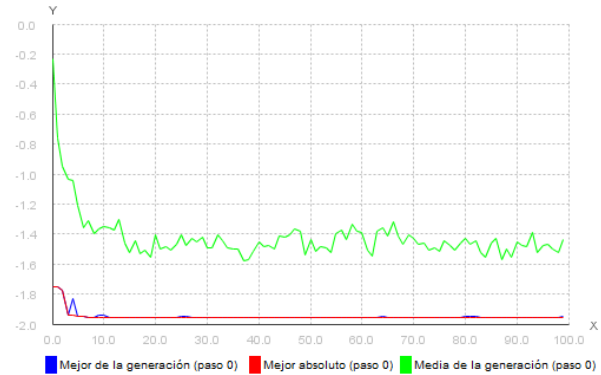
En todas ellas se puede ver como ha medida que la probabilidad de mutación aumenta los resultados son peores, en todos los sentidos. En particular, para $n = 5$, podemos ver como nos aproximamos al mínimo para el menor de este parámetro y que incluso la media para este caso mejora al mejor resultado obtenido para la ejecución con valores de probabilidad de mutación por encima del 5%. Todavía esta situación se agrava más en el caso $n = 7$ donde hay una diferencia de hasta 2 unidades entre la **media** de la ejecución con probabilidad 1% y el **mejor** del que tiene probabilidad 10%.

3.5.5. Elitismo



Sin Elitismo $f: -1.9573208915649871$ $x_0: 1.5648040314718776$ $x_1: 1.2826388293015611$

Elitismo $f: -1.959089286728108$ $x_0: 1.5706525117071441$ $x_1: 1.285131624155937$

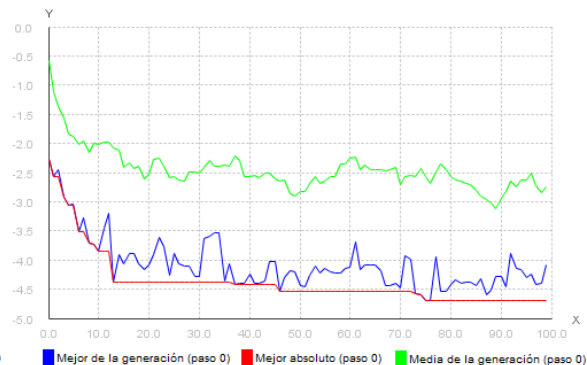


Sin Elitismo $f: -4.693464312444209$ $x_0: 1.5423688777824947$ $x_1: 1.2463015504627741$

$x_2: 1.129619575932949$ $x_3: 1.7124541882310647$ $x_4: 1.572474169485342$

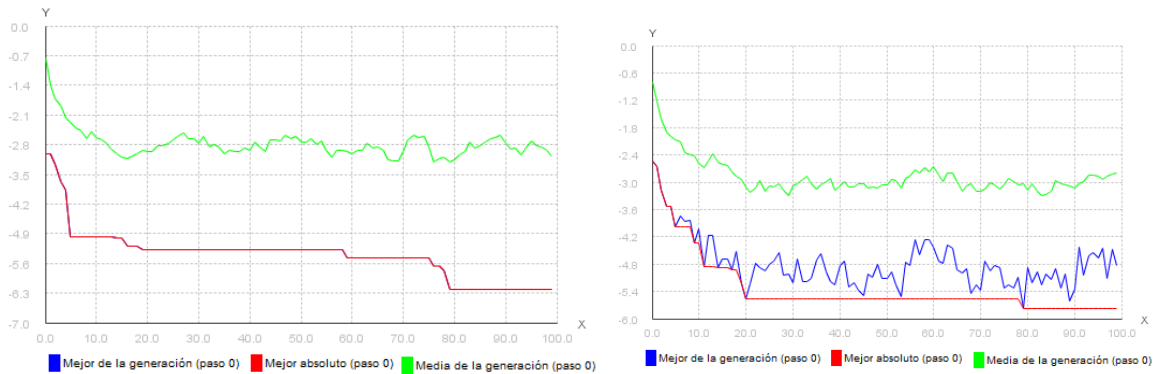
Elitismo $f: -4.825437445763266$ $x_0: 1.5726659229356783$ $x_1: 1.2798584042716803$

$x_2: 1.1213741775684751$ $x_3: 1.7184944219166678$ $x_4: 1.5765968686675789$



Programación Evolutiva - Práctica 1

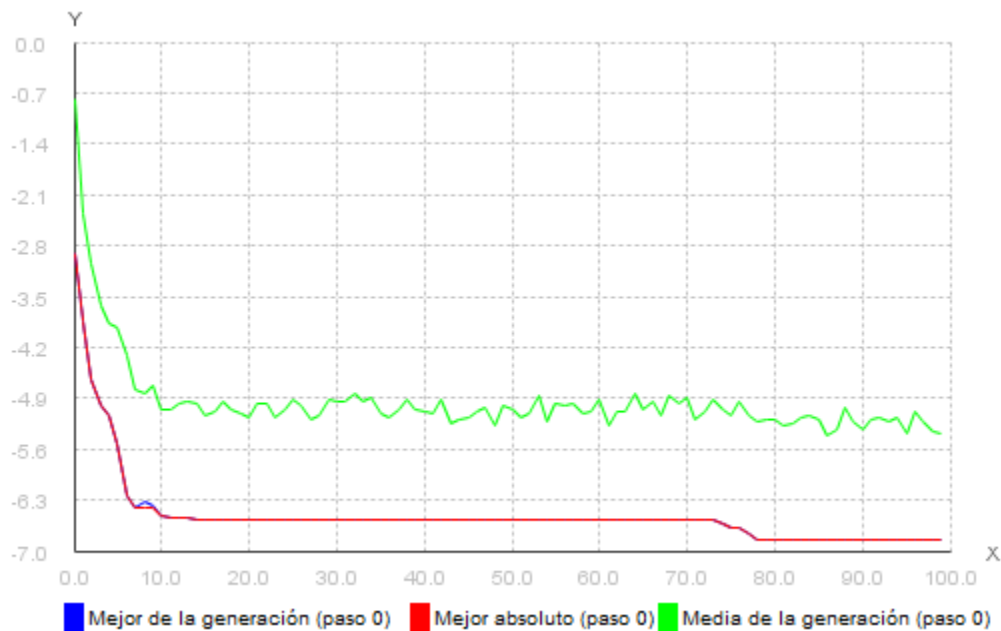
Algoritmo Genético Clásico



Sin Elitismo $f: -5.77410947798618$ $x0: 1.5820618420021721$ $x1: 1.3434246730582655$
 $x2: 1.0777502676168969$ $x3: 1.7061263243699565$ $x4: 1.5820618420021721$
 $x5: 0.8597265945841753$ $x6: 2.095961088904278$
Elitismo $f: -6.212015681652291$ $x0: 1.621850682947018$ $x1: 1.261450073039366$
 $x2: 1.942078945009151$ $x3: 1.7178232848404895$ $x4: 2.0241494217532185$
 $x5: 1.4265497937791844$ $x6: 1.350327797270383$

En todos los casos un elitismo del 2% consigue tener un efecto positivo sobre los resultados del mejor de cada generación, puesto que en definitiva lo estamos guardando como parte de esa élite.

3.5.6. Selección por ruleta y torneo

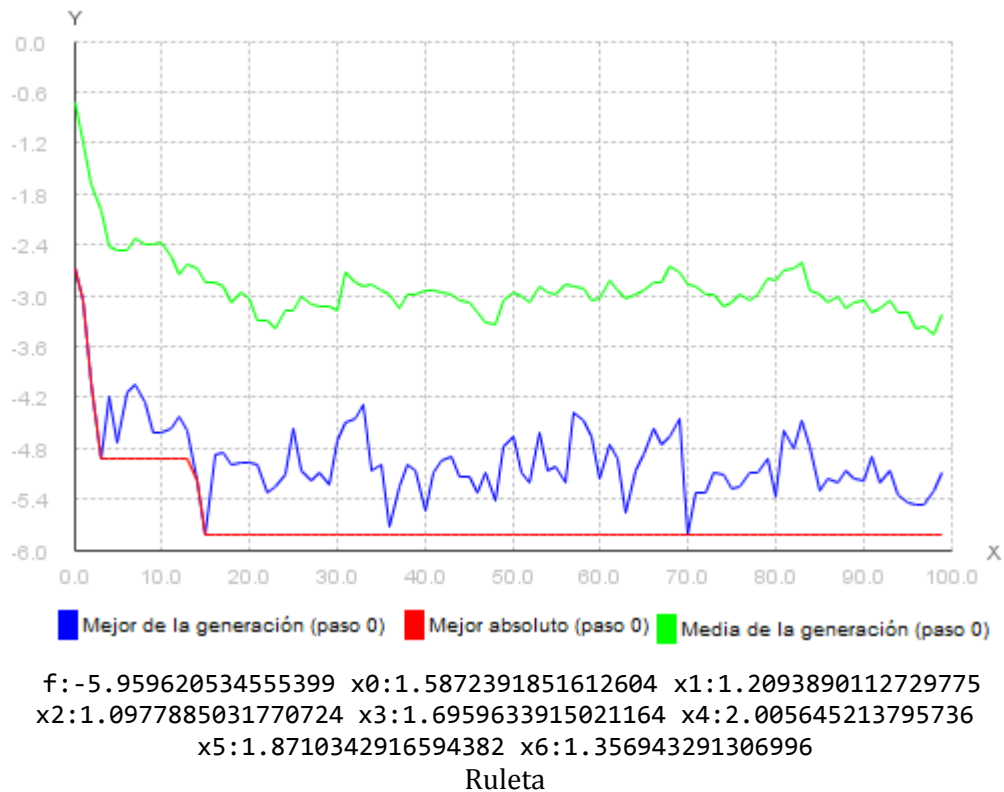


$f: -6.860500862400756$ $x0: 1.568734977203778$ $x1: 1.2826388293015611$
 $x2: 1.9223283396244806$ $x3: 1.7202202029696974$ $x4: 1.5708442651574808$
 $x5: 1.453970537177319$ $x6: 1.7548317007554546$

Torneo

Programación Evolutiva - Práctica 1

Algoritmo Genético Clásico



Al igual que en las anteriores funciones la selección por torneo ofrece mejores resultados que la selección por ruleta en términos de rapidez de convergencia así como a la hora de obtener el mejor resultado para el problema.

4. Conclusiones

Para finalizar, podemos concluir con que en general, no se puede fijar una conjunto de valores para los parámetros que pueda ser considerada óptimo para todos los problemas, debemos estudiar las características de la función que se desee optimizar y estadísticamente determinar la configuración óptima que nos proporcione un mejor comportamiento del algoritmo para nuestro problema.